# OS2复习

1. **带计算器**
2. 题型
   1. 判断20 1题1分

注意TF

* 1. 单选20 1题1分
  2. 填空20 1题1分
  3. 简答30 8 7 5

描述概念，会划重点，能描述

* 1. 计算3道20
  2. 填空选择很多与段/页有关

1. 第四章

都是填选

* 1. 程序的装入和链接 p131

装入——三道选择（和书上概念不同）

* + 1. 选择：**程序**装入内存才能运行，程序本质是指**可执行文件**
    2. 填空：装入本质是**建立地址空间映射**，把逻辑地址转化为物理地址的过程

即使本来就装在内存里，但这个时候还是逻辑地址，只有装入才转化成物理地址，建立一个地址空间，则建立一个地址映射。 把程序装入内存，其中“内存”是“地址空间”，即装入虚拟地址空间 **判断？：不是装入到物理地址就叫装入**

* + 1. 动态运行的装入方式 p133 4.2.1
    2. exec()

如果一个进程想执行另一个程序，它就可以fork出一个新进程，然后调用任何一个exec，这样看起来就好像通过执行应用程序而产生了一个新进程一样。

* + 1. 选择：读进来靠**缺页中断**，读满了靠**对换**
  1. 划内存
     1. 判断：只要是地址空间都可以用分区分配方式划分

分区分配方式，可以用在内存，也可以是虚拟空间，分配的是空间

* 1. 首次适应、最佳适应
     1. 判断：最佳适应不是真的最好的，最坏的不是真的最坏的
  2. 伙伴系统

读一读，可能是简答 每次只申请一个空闲空间，每次一分为二，有时候只需要一个块 以2^m去分配内存大小 分配时内需分配，长度为n，需要找到2^i，满足2^(i-1) < n < 2^i 是linux广泛使用的分区分配方式 回收的时候也一样，以2^m回收

* 1. 对换4.4

不太重要，后续要知道和虚存里对换的区别 这里的对换是指进程之间对换 虚存对换是进程内的对换

* 1. 分页4.5

分页是便于硬件管理，消减内存的外零头，便于内存管理；分页的大小由**硬件**（CPU）决定 分段是便于用户管理，由用户需求决定 分页=页号+页内偏移，**一维** 分段=段号+段长+基地址，**二维**

* + 1. 选择/填空：分页的大小、分页结构由**硬件**（CPU）决定
  1. 地址变换机构MMU p149

是**硬件**查找，因为操作太频繁，每一条指令都要查表 程序创建进程，所有都是逻辑地址，装入后变换为物理地址（查表） 页表的长度都放在PCB里 读一读，过程p150~151

* + 1. 选择/填空：地址变换是**硬件**查找
    2. 选择/填空：地址映射是**硬件**做的
  1. 快表
     1. 填空？：**快表TLB**是为了加快MMU的匹配过程（加快地址映射的速度），只能备份页表当前的活跃部分，是一种访问速度比内存快很多的**高速缓存**

快表（TLB）的地址变换系统，又称**联想寄存器** 是一种访问速度比内存快很多的高速缓存（TLB不是内存！）

* + 1. 如果不在TLB中，要去内存找匹配，然后放到快表——快表与内存的交换本质也是页面置换
  1. 多级页表
     1. 判断：不是为了提升速度，而是为了增加内存

目的：页表很大挤压内存，为**增加内存空间，降低所需存储器的容量**，引出二级页表

* 1. 反置页表
     1. 选择？：反置页表是整个内存一张表，把物理块作为表项，支持的也是**物理地址到逻辑地址**的映射和查询​

普通页表是一个进程一张表，造成内存浪费

* + 1. 填空？：前置页表根据**虚空间**设置页表，逆向页表根据**哈希值**设置页表（如果不能对应，则产生缺页中断）
  1. 分段
     1. 选择？：段式系统地址是二维的，页式是一维的

记录段长，一条指令有操作码和地址码，地址是虚地址，地址码里面需要有段长、段号和段内偏移，然后在和基地址匹配

1. 第五章

请求和置换算法

* 1. 虚存5.2
     1. 页表中的相关信息

页表做地址映射，所以一定有物理块号

* + 1. 重点：请求分页管理与前面的分页管理的区别
    2. “请求”是什么？

**“请求”是什么？**因为**缺页中断**，中断触发OS去把内容从外存调入内存。缺页中断的调入不是用系统调用，中断用软件中断。 中断处理完后，返回程序，继续执行指令 有虚存之后，程序越来越大，不需要全部调入内存才能运行

* + 1. 选择/填空？：请求页表的结构

状态位P只有1位，是指是否调入内存 访问字段A，被访问次数 修改位M，调入内存后是否修改过 ？（没有外存地址；地址映射映射到文件，不是设备——设备独立性）

* + 1. 虚存作用

虚拟存储、存储扩充使数据可以放到外存（文件），所以数据不用全部放到内存，只要当前执行的指令所需的数据在内存即可，不然则用缺页中断调入内存。但是也需要足够内存，否则缺页中断时间过长，运行效率反而较低

* 1. 综合题：页面置换算法5.3

大题，计算题，**细心** 判断：页面置换算法本质上是一个淘汰算法 **FIFO LRU OPT**会画图 题中已知需要5页分3页，形成置换背景，才要调来调去 最前面调入的过程也算不命中，算缺页率的时候记得算进去

* 1. 抖动 不考
  2. 分段5.5
     1. 判断：对Linux是段页式，对Windows不是段/页，不是段页这套系统的

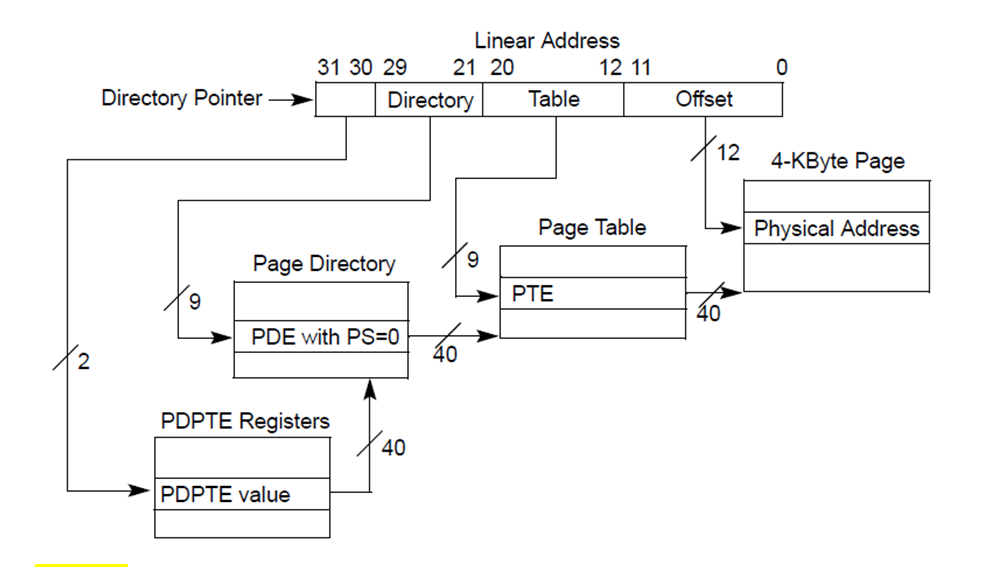
1. 第六章
   1. 常识性

虚存实现程序和物理空间是隔离的，实现至少一层物理上的抽象

* 1. I/O系统的层次结构和模型

用户层软件：printf,scanf ​设备独立性软件（就是文件系统）：mknod()，文件名和设备联系起来 ​设备驱动程序 ​中断处理程序 ​硬件

* 1. 简答题：PAE模式地址映射过程 p152~153 4-17大致一样



简答题 能描述地址映射过程，要小小总结一下 研讨题：奔腾处理器通过PAE模式实现超过4G物理内存的支持（可以映射到52位的物理地址），论述PAE模式地址映射过程 PDPTE Registers是CR3 用于保存 ​页目录是一个指针，指向下一级40位 借助了其他快速的硬件设备，40+12拼成52页地址 ​映射到xxx，得到物理地址xxx

* 1. 输入输出p194的图

五个空填全部五个 三个空就填中间三个 6-1的图

* 1. 设备驱动程序

属于I/O系统，不是操作系统开发 操作系统不开发任何I/O设备

* + 1. 操作系统不开发任何I/O设备，只负责页表维护、响应缺页中断等
  1. 假脱机p220

未过时 ​低速设备的输入输出 以磁盘为中介

* 1. 磁盘存储器p232
     1. 判断：磁盘调度是为了减少**寻道时间、旋转时间**√

大头是寻道时间和旋转时间，传输时间不是大头

* 1. 综合题：磁盘调度算法

**大题** FIFO SSTF 电梯算法 ​循环算法不用 lfy:fcfs?

1. 第七章 文件
   1. 索引节点inode

了解基本内容 FCB的内容本质是在inode里

* 1. 文件目录

不是给用户看的，可以呈现之前用户干的事情（打开关闭） 文件控制块FCB的属性，怎么引出索引节点 了解索引节点里文件标识符......

* 1. 文件的结构
     1. 判断：OS就看做是都是无结构的，文件都是一个字节一个字节的数据

excel也不是有结构的

* 1. unix系统的文件类型

**填空四种：正规文件、目录文件、块设备文件、字符设备文件** 块设备文件上**构建设备文件抽象**

* 1. 简答题：文件共享p257

**概念描述** 软链接与硬链接的区别，自己总结 硬链接p258 图7-13：利用索引节点，多个目录同时指向一个inode，加一个count 软链接/符号链接：有/s就是软链接，创建一个链接文件，把路径拷贝过来，其中的路径指向inode，一个文件/子目录有一个主父目录和多个父目录 ​优点：不会出现删除一个共享文件，留下悬空指针 缺点：每次访问开销大，耗费磁盘空间

* + 1. lfy？：文件系统的几种数据结构
  1. 简答题：实验-文件链接
     1. 文件链接

link()链接-**文件换名** unlink()断链接

* + 1. 改变文件

chmod()改变文件权限 ​choun()改变文件所有者

* + 1. 文件安装卸载

Linux中用mount, unmount安装卸下其他盘 对比Windows系统中读写U盘 mount()加载文件系统，实现多文件系统 unmount()卸载多文件系统 ​Mount umount既是系统功能调用，也是命令

* + 1. 文件打开

函数系统调用 linux文件使用前要打开 linux访问文件open() 读read()写write() open的返回值的是**文件描述符fd** 之后操作全通过fd，使用完毕要关闭文件

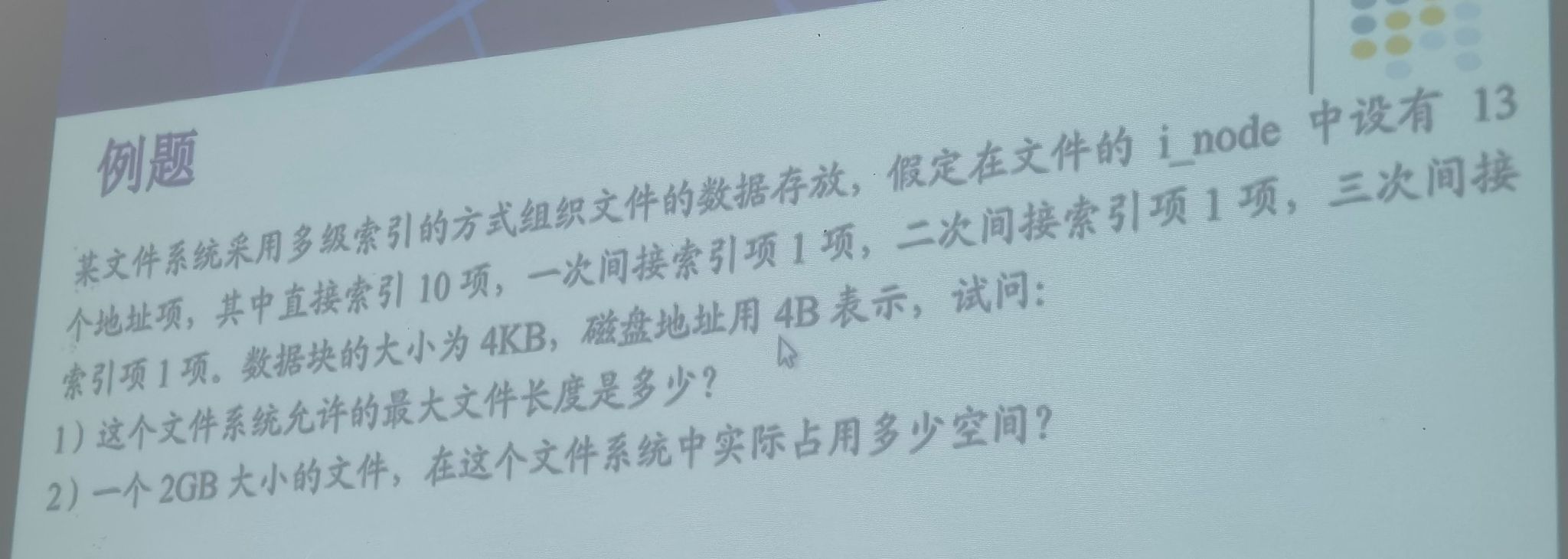
1. 第八章

大题

* 1. 综合题：磁盘管理

管理磁盘块，索引离散的 unix用的是图8-8，**混合式索引/增量式索引**，是**固定的** 13个地址项怎么分配？0~12里0~9是10个直接地址（直接指向块data），1个一次间接，1个二次间接，一个三次间接

* + 1. 1 文件xxMB，会占多少磁盘空间/索引块
    2. 2 访问次数，读写多少磁盘
    3. 3 读写哪个盘块
    4. 特点：小文件很快，大文件也可以（有页面置换算法支持）
  1. 例题



eg. 一个盘块4K，如果存的是地址，每个表项4B，当文件不大于40K的时候，就是直接地址就行了（10个直接地址\*4），文件大就得用上间接地址，有1K个盘块号作为索引 每个数据块大小是4KB ①10×4KB（直接）+ 4KB/4B（1KB个表项）×4KB（一个表项4KB）（一级间接）+1KB（1KB个表项）×1KB×4KB（二级间接）+1KB×1KB×1KB×4KB（三级间接） 大约就是4TB ②2GB/4KB = 512×1024块 要算索引（主要都在二级（4GB），三级就没有索引了） 二级索引需要的块数 **1**+(512×1024-10-1024)/1024=512 13×4B = 52B 实际占用空间：2GB(数据) + 52B（inode） + (1(一级)+512(二级))×4KB

* + 1. 第三问 求物理盘块号

若 文件大小是12450000B，要求转换为物理盘块号和块内偏移 12450000/4096(4KB) 12450000 = 4096×3039 + 2256 3039-10（直接） - 1024（一次的表项） = 2005 2005 = 1024×**1**（一级索引项） + **981**（二级索引项） 答案是1和981，是二级间接寻址 ​

* + 1. 访问磁盘数

三级间接寻址访问四次磁盘 ？一级是一次吗

* 1. 文件存储空间管理 8.2 p278

磁盘文件区的管理想：混合索引，要分盘块，文件存储空间管理就是如何把空闲盘块放在一起 线性链表的方式分盘块（不重要） **位示图法和成组链接法**（重要 p281 ***简答题***） 利用二进制的1位来表示磁盘中的一个盘块使用情况：0为空闲，1为已分配

* + 1. 位示图法

**位示图**：磁盘上的所有盘块都有一个二进制位与之对应，这样由所有盘块所对应的位构成一个集合 **优点**：大小不大 **缺点**：遍历搜索复杂（需要补充）

* + 1. 简答题：成组链接法

**成组链接法**：结合了空闲表法和空闲链表法而形成的一种空闲盘块管理方法，克服表太长的缺点（**UNIX采用**） ​空闲盘块号栈——超级块，每100个为一组 将第一组的盘块总数和所有的盘块号，记入空闲盘块号栈 每组的盘块数和盘块号记入前一组

* + - 1. 读写了几次磁盘，第一次写读多少磁盘
    1. 网上能找到的例题